Network Working Group
Request for Comments: 5444
Category: Standards Track

T. Clausen
LIX, Ecole Polytechnique
C. Dearlove
BAE Systems ATC
J. Dean
Naval Research Laboratory
C. Adjih
INRIA Rocquencourt

February 2009

Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format

Status of This Memo

Copyright Notice

Abstract

This document specifies a packet format capable of carrying multiple messages that may be used by mobile ad hoc network routing protocols.

Table of Contents

1. Introduction

This document specifies the syntax of a packet format designed for carrying multiple routing protocol messages for information exchange between MANET (Mobile Ad hoc NETwork) routers. Messages consist of a Message Header, which is designed for control of message dissemination, and a Message Body, which contains protocol information. Only the syntax of the packet and messages is specified.

This document specifies:

- A packet format, allowing zero or more messages to be contained within a single transmission. A packet with zero messages may be sent in case the only information to exchange is contained in the Packet Header.

- A message format, where a message is composed of a Message Header and a Message Body.

- A Message Header format, which contains information that may be sufficient to allow a protocol using this specification to make processing and forwarding decisions.

- A Message Body format, containing attributes associated with the message or the originator of the message, as well as blocks of addresses, or address prefixes, with associated attributes.

- An Address Block format, where an Address Block represents sets of addresses, or address prefixes, in a compact form with aggregated addresses.

- A generalized type-length-value (TLV) format representing attributes. Each TLV can be associated with a packet, a message, or one or more addresses or address prefixes in a single Address Block. Multiple TLVs can be included, each associated with a packet, a message, and the same, different, or overlapping sets of addresses or address prefixes.

The specification has been explicitly designed with the following properties, listed in no particular order, in mind:

Parsing logic - The notation used in this specification facilitates generic, protocol-independent parsing logic.

Extensibility - Packets and messages defined by a protocol using this specification are extensible by defining new messages and new TLVs. Protocols using this specification will be able to correctly identify and skip such new messages and TLVs, while correctly parsing the remainder of the packet and message.

Efficiency - When reported addresses share common bit sequences (e.g., address prefixes or IPv6 interface identifiers), the Address Block representation allows for a compact representation. Compact Message Headers are ensured through permitting inclusion of only required Message Header elements. The multi-message packet structure allows a reduction in the number of transmitted octets and in the number of transmitted packets. The structure of packet and message encoding allows parsing, verifying, and identifying individual elements in a single pass.

Separation of forwarding and processing - A protocol using this specification can be designed such that duplicate detection and controlled-scope message forwarding decisions can be made using information contained in the Message Header, without processing the Message Body.

2. Notation and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, this document uses the notation in Section 2.1 and the terminology in Section 2.2.

2.1. Notation

The following notations, for elements and variables, are used in this document.

This format uses network byte order (most significant octet first) for all fields. The most significant bit in an octet is numbered bit 0, and the least significant bit of an octet is numbered bit 7 [Stevens].

2.1.1. Elements

This specification defines elements. An element is a group of any number of consecutive bits that together form a syntactic entity represented using the notation <element>. Each element in this document is defined as either:

o a specifically sized field of bits OR

o a composite element, composed of other <element>s.

A composite element is defined as follows:

<element> := specification

where, on the right hand side following :=, specification is represented using the regular expression syntax defined in [SingleUNIX]. Only the following notation is used:

<element1><element2> - Indicates that <element1> is immediately followed by <element2>.

(<element1><element2>) - Indicates a grouping of the elements enclosed by the parentheses.

? - Zero or one occurrences of the preceding element or group.

* - Zero or more occurrences of the preceding element or group.

2.1.2. Variables

Variables are introduced into the specification solely as a means to clarify the description. The following two notations are used:

<foo> - If <foo> is an unsigned integer field, then <foo> is also used to represent the value of that field.

bar - A variable, usually obtained through calculations based on the value(s) of element(s).

2.2. Terminology

This document uses the following terminology:

Packet - The top level entity in this specification. A packet contains a Packet Header and zero or more messages.

Message - The fundamental entity carrying protocol information, in the form of address objects and TLVs.

Address - A number of octets that make up an address of the length indicated by the encapsulating Message Header. The meaning of an address is defined by the protocol using this specification.

Address Prefix - An address plus a prefix length, with the prefix length being a number of address bits measured from the left/most significant end of the address.

Address Object - Either an address, or an address prefix, as specified in an Address Block in this specification.

TLV - A type-length-value structure. This is a generic way in which an attribute can be represented and correctly parsed without the parser having to understand the attribute.

3. Applicability Statement

This specification describes a generic packet format, designed for use by MANET routing protocols. The specification has been inspired by and extended from that used by the OLSR (Optimized Link State Routing) protocol [RFC3626].

MANETs are, commonly though not exclusively, characterized as being able to run over wireless network interfaces of limited to moderate capacity. MANETs are therefore less tolerant of wasted transmitted octets than are most wired networks. This specification thus represents a tradeoff between sometimes competing attributes, specifically efficiency, extensibility, and ease of use.

Efficiency is supported by reducing packet size and by allowing multiple disjoint messages in a single packet. Reduced packet size is primarily supported by address aggregation, optional Packet Header and Message Header fields, and optional fields in Address Blocks and TLVs. Supporting multi-message packets allows a reduction in the number of packets, each of which can incur significant bandwidth costs from transport, network, and lower layers.

This specification provides both external and internal extensibility. External extensibility is supported by the ability to add Packet TLVs and to define new Message Types. Internal extensibility is supported by the ability to add Message TLVs and Address Block TLVs to existing messages. Protocols can define new TLV Types, and hence the contents of their Value fields, and new Message Types (see Section 6.1). Protocols can also reuse TLV Type definitions from other protocols that also use this specification.

This specification aims at being sufficiently expressive and flexible to be able to accommodate different classes of MANET routing protocols (e.g., proactive, reactive, and hybrid routing protocols) as well as extensions thereto. Having a common packet and message

format, and a common way of representing IP addresses and associated attributes, allows generic parsing code to be developed, regardless of the algorithm used by the routing protocol.

All addresses within a message are assumed to be of the same size, specified in the Message Header. In the case of mixed IPv6 and IPv4 addresses, IPv4 addresses can be represented as IPv4-mapped IPv6 addresses as specified in [RFC4291].

The messages defined by this specification are designed to carry MANET routing protocol signaling between MANET routers. This specification includes elements that can support scope-limited flooding, as well as being usable for point-to-point delivery of MANET routing protocol signaling in a multi-hop network. Packets may be unicast or multicast and may use any appropriate transport protocol or none.

A MANET routing protocol using the message format defined by this specification can constrain the syntax (for example, requiring a specific set of Message Header fields) that the protocol will employ. Protocols with such restrictions need not be able to parse all possible message structures as defined by this document but must be coherent in message generation and reception of messages that they define. If a protocol specifies which elements are included, then direct indexing of the appropriate fields is possible, dependent on the syntax restrictions imposed by the protocol. Such protocols may have more limited extensibility.

4. Protocol Overview and Functioning

This specification does not describe a protocol. It describes a packet format, which may be used by any mobile ad hoc network routing protocol.

5. Syntactical Specification

This section normatively provides the syntactical specification of a packet, represented by the element <packet> and the elements from which it is composed. The specification is given using the notation in Section 2.1.

Graphical illustrations of the layout of specified elements are given in Appendix D, a graphical illustration of a complete example (a packet including a message with Address Blocks and TLVs) is given in Appendix E.

This format uses network byte order, as indicated in Section 2.1.

5.1. Packets

<packet> is defined by:

```
<packet> := <pkt-header>
            <message>*
```

where <message> is as defined in Section 5.2. Successful parsing is terminated when all octets of the packet (as defined by the datagram containing the packet) are used.

<pkt-header> is defined by:

```
<pkt-header> := <version>
                <pkt-flags>
                <pkt-seq-num>?
                <tlv-block>?
```

where:

<version> is a 4-bit unsigned integer field and specifies the version of the specification on which the packet and the contained messages are constructed. This document specifies version 0.

<pkt-flags> is a 4-bit field, specifying the interpretation of the remainder of the Packet Header:

bit 0 (phasseqnum): If cleared ('0'), then <pkt-seq-num> is not included in the <pkt-header>. If set ('1'), then <pkt-seq-num> is included in the <pkt-header>.

bit 1 (phastlv): If cleared ('0'), then <tlv-block> is not included in the <pkt-header>. If set ('1'), then <tlv-block> is included in the <pkt-header>.

bits 2-3: Are RESERVED and SHOULD each be cleared ('0') on transmission and SHOULD be ignored on reception.

<pkt-seq-num> is omitted if the phasseqnum flag is cleared ('0'); otherwise, is a 16-bit unsigned integer field, specifying a Packet Sequence Number.

<tlv-block> is omitted if the phastlv flag is cleared ('0') and is otherwise as defined in Section 5.4.

It is assumed that the network layer is able to deliver the exact payload length, thus avoiding having to carry the packet length in the packet.

5.2. Messages

Packets may, in addition to the Packet Header, contain one or more messages. Messages contain:

- o A Message Header.
- o A Message TLV Block that contains zero or more TLVs, associated with the whole message.
- o Zero or more Address Blocks, each containing one or more address objects.
- o An Address Block TLV Block, containing zero or more TLVs and following each Address Block, through which addresses can be associated with additional attributes.

<message> is defined by:

```
<message>    := <msg-header>
                <tlv-block>
                (<addr-block><tlv-block>)*

<msg-header> := <msg-type>
                <msg-flags>
                <msg-addr-length>
                <msg-size>
                <msg-orig-addr>?
                <msg-hop-limit>?
                <msg-hop-count>?
                <msg-seq-num>?
```

where:

<tlv-block> is as defined in Section 5.4.

<addr-block> is as defined in Section 5.3.

<msg-type> is an 8-bit unsigned integer field, specifying the type of the message.

<msg-flags> is a 4-bit field, specifying the interpretation of the remainder of the Message Header:

bit 0 (mhasorig): If cleared ('0'), then <msg-orig-addr> is not included in the <msg-header>. If set ('1'), then <msg-orig-addr> is included in the <msg-header>.

bit 1 (mhashoplimit): If cleared ('0'), then <msg-hop-limit> is not included in the <msg-header>. If set ('1'), then <msg-hop-limit> is included in the <msg-header>.

bit 2 (mhashopcount): If cleared ('0'), then <msg-hop-count> is not included in the <msg-header>. If set ('1'), then <msg-hop-count> is included in the <msg-header>.

bit 3 (mhasseqnum): If cleared ('0'), then <msg-seq-num> is not included in the <msg-header>. If set ('1'), then <msg-seq-num> is included in the <msg-header>.

<msg-addr-length> is a 4-bit unsigned integer field, encoding the length of all addresses included in this message (<msg-orig-addr> as well as each address included in Address Blocks as defined in Section 5.3), as follows:

<msg-addr-length> = the length of an address in octets - 1

<msg-addr-length> is thus 3 for IPv4 addresses, or 15 for IPv6 addresses.

address-length is a variable whose value is the length of an address in octets and is calculated as follows:

address-length = <msg-addr-length> + 1

<msg-size> is a 16-bit unsigned integer field, specifying the number of octets that make up the <message>, including the <msg-header>.

<msg-orig-addr> is omitted if the mhasorig flag is cleared ('0'); otherwise, is an identifier with length equal to address-length that can serve to uniquely identify the MANET router that originated the message.

<msg-hop-limit> is omitted if the mhashoplimit flag is cleared ('0'); otherwise, is an 8-bit unsigned integer field that can contain the maximum number of hops that the message should be further transmitted.

<msg-hop-count> is omitted if the mhashopcount flag is cleared ('0'); otherwise, is an 8-bit unsigned integer field that can contain the number of hops that the message has traveled.

<msg-seq-num> is omitted if the mhasseqnum flag is cleared ('0'); otherwise, is a 16-bit unsigned integer field that can contain a sequence number, generated by the message's originator MANET router.

5.3. Address Blocks

An Address Block can specify one or more addresses, all of which will be address-length octets long, as specified using the <msg-addr-length> in the <msg-header> of the message containing the Address Block. An Address Block can also specify prefix lengths that can be applied to all addresses in the Address Block, if appropriate. This allows an Address Block to specify either addresses or address prefixes. A protocol may specify that an address with a maximum prefix length (equal to the address length in bits, i.e., 8 * address-length) is considered to be an address, rather than an address prefix, thus allowing an Address Block to contain a mixture of addresses and address prefixes. The common term "address object" is used in this specification to cover both of these; note that an address object in an Address Block always includes the prefix length, if present.

An address is specified as a sequence of address-length octets of the form Head:Mid:Tail. There are no semantics associated with Head, Mid, or Tail; this representation is solely to allow aggregation of addresses, which often have common parts (e.g., common prefixes or multiple IPv6 addresses on the same interface). An Address Block contains an ordered set of addresses all sharing the same Head and the same Tail, but having individual Mids. Independently, Head and Tail may be empty, allowing for representation of address objects that do not have common Heads or common Tails. Detailed examples of Address Blocks are included in Appendix C.1.

An Address Block can specify address prefixes:

- o with a single prefix length for all address prefixes OR
- o with a prefix length for each address prefix.

<address-block> is defined by:

```
<address-block> := <num-addr>
                   <addr-flags>
                   (<head-length><head>?)?
                   (<tail-length><tail>?)?
                   <mid>*
                   <prefix-length>*
```

where:

<num-addr> is an 8-bit unsigned integer field containing the number of addresses represented in the Address Block, which MUST NOT be zero.

<addr-flags> is an 8-bit field specifying the interpretation of the remainder of the Address Block:

bit 0 (ahashead): If cleared ('0'), then <head-length> and <head> are not included in the <address-block>. If set ('1'), then <head-length> is included in the <address-block>, and <head> is included in the <address-block> unless <head-length> is zero.

bit 1 (ahasfulltail) and bit 2 (ahaszerotail): Are interpreted according to Table 1. A combination not shown in that table MUST NOT be used.

bit 3 (ahassingleprelen) and bit 4 (ahasmultiprelen): Are interpreted according to Table 2. A combination not shown in that table MUST NOT be used.

bits 5-7: Are RESERVED and SHOULD each be cleared ('0') on transmission and SHOULD be ignored on reception.

ahasfulltail	ahaszerotail	<tail-length>	<tail>
0	0	not included	not included
1	0	included	included unless <tail-length> is zero
0	1	included	not included

Table 1: Interpretation of the ahasfulltail and ahaszerotail flags

ahassingle prelen	ahasmulti prelen	number of <prefix-length> fields	prefix length of the nth address prefix, in bits
0	0	0	8 * address-length
1	0	1	<prefix-length>
0	1	<num-addr>	nth <prefix-length>

Table 2: Interpretation of the ahassingleprelen and ahasmultiprelen flags

<head-length> if present, is an 8-bit unsigned integer field that contains the number of octets in the Head of all of the addresses in the Address Block, i.e., each <head> element included is <head-length> octets long.

head-length is a variable, defined to equal <head-length>, if present, or 0 otherwise.

<head> is omitted if head-length is equal to 0; otherwise, it is a field of the head-length leftmost octets common to all the addresses in the Address Block.

<tail-length> if present, is an 8-bit unsigned integer field that contains the number of octets in the Tail of all of the addresses in the Address Block, i.e., each <tail> element included is <tail-length> octets long.

tail-length is a variable, defined to equal <tail-length>, if present, or 0 otherwise.

<tail> is omitted if tail-length is equal to 0, or if the ahaszerotail flag is set ('1'); otherwise, it is a field of the tail-length rightmost octets common to all the addresses in the Address Block. If the ahaszerotail flag is set ('1'), then the tail-length rightmost octets of all the addresses in the Address Block are 0.

mid-length is a variable that MUST be non-negative, defined by:

    mid-length := address-length - head-length - tail-length

i.e., each <mid> element included is mid-length octets long.

<mid> is omitted if mid-length is equal to 0; otherwise, each <mid> is a field of length mid-length octets, representing the Mid of the corresponding address in the Address Block. When not omitted, an Address Block contains exactly <num-addr> <mid> fields.

<prefix-length> is an 8-bit unsigned integer field containing the length, in bits, of an address prefix. If the ahassingleprelen flag is set ('1'), then a single <prefix-length> field is included that contains the prefix length of all addresses in the Address Block. If the ahasmultiprelen flag is set ('1'), then <num-addr> <prefix-length> fields are included, each of which contains the prefix length of the corresponding address prefix in the Address Block (in the same order). Otherwise, no <prefix-length> fields are present; each address object can be considered to have a prefix length equal to 8 * address-length bits. The Address Block is malformed if any <prefix-length> element has a value greater than 8 * address-length.

5.4. TLVs and TLV Blocks

A TLV allows the association of an arbitrary attribute with a message or a packet, or with a single address or a contiguous set of addresses in an Address Block. The attribute (value) is made up from an integer number of consecutive octets. Different attributes have different types; attributes that are unknown when parsing can be skipped.

TLVs are grouped in TLV Blocks, with all TLVs within a TLV Block associating attributes with either the packet (for the TLV Block in the Packet Header), the message (for the TLV Block immediately following the Message Header), or to addresses in the immediately preceding Address Block. Individual TLVs in a TLV Block immediately following an Address Block can associate attributes to a single address, a range of addresses, or all addresses in that Address Block. When associating an attribute with more than one address, a TLV can include one value for all addresses or one value per address. Detailed examples of TLVs are included in Appendix C.2.

A TLV Block is defined by:

```
    <tlv-block> := <tlvs-length>
                   <tlv>*
```

where:

<tlvs-length> is a 16-bit unsigned integer field that contains the total number of octets in all of the immediately following <tlv> elements (<tlvs-length> not included).

<tlv> is as defined in Section 5.4.1.

5.4.1. TLVs

There are three kinds of TLV, each represented by an element <tlv>:

- o A Packet TLV, included in the Packet TLV Block in a Packet Header.
- o A Message TLV, included in the Message TLV Block in a message, before any Address Blocks.
- o An Address Block TLV, included in an Address Block TLV Block following an Address Block. An Address Block TLV applies to:

* all address objects in the Address Block, OR

* any continuous sequence of address objects in the Address Block, OR

* a single address object in the Address Block.

<tlv> is defined by:

```
<tlv> := <tlv-type>
         <tlv-flags>
         <tlv-type-ext>?
         (<index-start><index-stop>?)?
         (<length><value>?)?
```

where:

<tlv-type> is an 8-bit unsigned integer field, specifying the type of the TLV, specific to the TLV kind (i.e., Packet TLV, Message TLV, or Address Block TLV).

<tlv-flags> is an 8-bit field specifying the interpretation of the remainder of the TLV:

bit 0 (thastypeext): If cleared ('0'), then <tlv-type-ext> is not included in the <tlv>. If set ('1'), then <tlv-type-ext> is included in the <tlv>.

bit 1 (thassingleindex) and bit 2 (thasmultiindex): Are interpreted according to Table 3. A combination not shown in that table MUST NOT be used. Both of these flags MUST be cleared ('0') in Packet TLVs and Message TLVs.

bit 3 (thasvalue) and bit 4 (thasextlen): Are interpreted according to Table 4. A combination not shown in that table MUST NOT be used.

bit 5 (tismultivalue): This flag serves to specify how the <value> field is interpreted, as specified below. This flag MUST be cleared ('0') in Packet TLVs and Message TLVs, if the thasmultiindex flag is cleared ('0'), or if the thasvalue flag is cleared ('0').

bits 6-7: Are RESERVED and SHOULD each be cleared ('0') on transmission and SHOULD be ignored on reception.

thassingleindex	thasmultiindex	<index-start>	<index-stop>
0	0	not included	not included
1	0	included	not included
0	1	included	included

Table 3: Interpretation of the thassingleindex and thasmultiindex flags

thasvalue	thasextlen	<length>	<value>
0	0	not included	not included
1	0	8 bits	included unless <length> is zero
1	1	16 bits	included unless <length> is zero

Table 4: Interpretation of the thasvalue and thasextlen flags

<tlv-type-ext> is an 8-bit unsigned integer field, specifying an extension of the TLV Type, specific to the TLV Type and kind (i.e., Packet TLV, Message TLV, or Address Block TLV).

tlv-type-ext is a variable, defined to equal <tlv-type-ext>, if present, or 0 otherwise.

tlv-fulltype is a variable, defined by:

    tlv-fulltype := 256 * <tlv-type> + tlv-type-ext

<index-start> and <index-stop> when present, in an Address Block TLV only, are each an 8-bit unsigned integer field.

index-start and index-stop are variables, defined according to Table 5. The variable end-index is defined as follows:

* For Message TLVs and Packet TLVs:

    end-index := 0

* For Address Block TLVs:

    end-index := <num-addr> - 1

An Address Block TLV applies to the address objects from position index-start to position index-stop (inclusive) in the Address Block, where the first address object has position zero.

thassingleindex	thasmultiindex	index-start :=	index-stop :=
0	0	0	end-index
1	0	<index-start>	<index-start>
0	1	<index-start>	<index-stop>

Table 5: Interpretation of the thassingleindex and thasmultiindex flags

number-values is a variable, defined by:

number-values := index-stop - index-start + 1

<length> is omitted or is an 8-bit or 16-bit unsigned integer field according to Table 4. If the tismultivalue flag is set ('1'), then <length> MUST be an integral multiple of number-values, and the variable single-length is defined by:

single-length := <length> / number-values

If the tismultivalue flag is cleared ('0'), then the variable single-length is defined by:

single-length := <length>

<value> if present (see Table 4), is a field of length <length> octets. In an Address Block TLV, <value> is associated with the address objects from positions index-start to index-stop, inclusive. If the tismultivalue flag is cleared ('0'), then the whole of this field is associated with each of the indicated address objects. If the tismultivalue flag is set ('1'), then this field is divided equally into number-values fields, each of length single-length octets, and these are associated, in order, with the indicated address objects.

5.4.2. TLV Usage

A TLV associates an attribute with a packet, a message, or one or more consecutive address objects in an Address Block. The interpretation and processing of this attribute, and the relationship

(including order of processing) between different attributes associated with the same entity MUST be defined by any protocol that uses this specification.

Any protocol using this specification MUST define appropriate behaviors if this associated information is inconsistent, in particular if two TLVs of the same type but with different values apply to the same entity (packet, message, or address) but this is not meaningful. The protocol MUST also specify an appropriate processing order for TLVs associated with a given entity.

5.5. Malformed Elements

An element is malformed if it cannot be parsed according to its syntactical specification (including if there are insufficient octets available). If the malformed element is in the Packet Header, then the packet MUST be silently discarded, and contained messages MUST NOT be processed and MUST NOT be forwarded. If the malformed element is contained in a message (i.e., is in the Message TLV Block, an Address Block, or an Address Block TLV Block), then that message MUST be silently discarded; it MUST NOT be processed and MUST NOT be forwarded.

6. IANA Considerations

This document introduces four namespaces that have been registered: Message Types, Packet TLV Types, Message TLV Types, and Address Block TLV Types. This section specifies IANA registries for these namespaces and provides guidance to the Internet Assigned Numbers Authority regarding registrations in these namespaces.

The following terms are used with the meanings defined in [BCP26]: "Namespace", "Assigned Value", "Registration", "Unassigned", "Reserved", "Hierarchical Allocation", and "Designated Expert".

The following policies are used with the meanings defined in [BCP26]: "Private Use", "Expert Review", and "Standards Action".

6.1. Expert Review: Evaluation Guidelines

For registration requests where an Expert Review is required, the Designated Expert SHOULD take the following general recommendations into consideration:

- The purpose of these registries is to support Standard and Experimental MANET routing and related protocols and extensions to these protocols.

- o The intention is that all registrations will be accompanied by a published RFC.

- o In order to allow for registration prior to the RFC being approved for publication, the Designated Expert can approve the registration once it seems clear that an RFC is expected to be published.

- o The Designated Expert will post a request to the MANET WG mailing list, or to a successor thereto as designated by the Area Director, for comments and reviews. This request will include a reference to the Internet-Draft requesting the registration.

- o Before a period of 30 days has passed, the Designated Expert will either approve or deny the registration request and publish a note of the decision to the MANET WG mailing list or its successor, as well as inform IANA and the IESG. A denial note MUST be justified by an explanation and, in cases where it is possible, suggestions as to how the request can be modified so as to become acceptable SHOULD be provided.

For the registry for Message Types, the following guidelines apply:

- o Registration of a Message Type implies creation of two registries for Message-Type-specific Message TLVs and Message-Type-specific Address Block TLVs. The document that requests the registration of the Message Type MUST indicate how these Message-Type-specific TLV Types are to be allocated, from any options in [BCP26], and any initial allocations. The Designated Expert SHOULD take the allocation policies specified for these registries into consideration in reviewing the Message Type allocation request.

For the registries for Packet TLV Types, Message TLV Types, and Address Block TLV Types, the following guidelines apply:

- o These are Hierarchical Allocations, i.e., allocation of a type creates a registry for the extended types corresponding to that type. The document that requests the registration of the type MUST indicate how these extended types are to be allocated, from any options in [BCP26], and any initial allocations. Normally this allocation should also undergo Expert Review, but with the possible allocation of some type extensions as Reserved, Experimental, and/or Private.

- o The request for a TLV Type MUST include the specification of the permitted size, syntax of any internal structure, and meaning, of the Value field (if any) of the TLV.

For the registries for Message TLV Types and Address Block TLV Types, the following additional guidelines apply:

- o TLV Type values 0-127 are common for all Message Types. TLVs that receive registrations from the 0-127 interval SHOULD be modular in design to allow reuse among protocols.

- o TLV Type values 128-223 are Message-Type-specific TLV Type values, relevant only in the context of the containing Message Type. Registration of TLV Type values within the 128-223 interval requires that a registry in the 128-223 interval exists for a specific Message Type value (see Section 6.2.1), and registrations are made in accordance with the allocation policies specified for these Message-Type-specific registries. Message-Type-specific TLV Types SHOULD be registered for TLVs that the Designated Expert deems too Message-Type-specific for registration of a 0-127 value. Multiple different TLV definitions MAY be assigned the same TLV Type value within the 128-223 interval, given that they are associated with different Message-Type-specific TLV Type registries. Where possible, existing global TLV definitions and modular global TLV definitions for registration in the 0-127 range SHOULD be used.

6.2. Message Types

A new registry for Message Types has been created, with initial assignments and allocation policies as specified in Table 6.

Type	Description	Allocation Policy
0-223	Unassigned	Expert Review
224-255	Unassigned	Experimental Use

Table 6: Message Types

6.2.1. Message-Type-Specific TLV Registry Creation

When a Message Type is registered, then registries MUST be specified for both Message-Type-specific Message TLVs (Table 8) and Message-Type-specific Address Block TLVs (Table 10). A document that creates a Message-Type-specific TLV registry MUST also specify the mechanism by which Message-Type-specific TLV Types are allocated, from among those in [BCP26].

6.3. Packet TLV Types

A new registry for Packet TLV Types has been created, with initial assignments and allocation policies as specified in Table 7.

Type	Description	Allocation Policy
0-223	Unassigned	Expert Review
224-255	Unassigned	Experimental Use

Table 7: Packet TLV Types

6.3.1. Packet TLV Type Extension Registry Creation

When a Packet TLV Type is registered, then a new registry for type extensions of that type must be created. A document that defines a Packet TLV Type MUST also specify the mechanism by which its type extensions are allocated, from among those in [BCP26].

6.4. Message TLV Types

A new registry for Message-Type-independent Message TLV Types has been created, with initial assignments and allocation policies as specified in Table 8.

Type	Description	Allocation Policy
0-127	Unassigned	Expert Review
128-223	Message-Type-specific	Reserved, see Table 9
224-255	Unassigned	Experimental Use

Table 8: Message TLV Types

Message TLV Types 128-223 are reserved for Message-Type-specific Message TLVs, for which a new registry is created with the registration of a Message Type, and with initial assignments and allocation policies as specified in Table 9.

Type	Description	Allocation Policy
0-127	Common to all Message Types	Reserved
128-223	Message-Type-specific	See Below
224-255	Common to all Message Types	Reserved

Table 9: Message-Type-specific Message TLV Types

Allocation policies for Message-Type-specific Message TLV Types MUST be specified when creating the registry associated with the containing Message Type, see Section 6.2.1.

6.4.1. Message TLV Type Extension Registry Creation

If a Message TLV Type is registered, then a new registry for type extensions of that type must be created. A document that defines a Message TLV Type MUST also specify the mechanism by which its type extensions are allocated, from among those in [BCP26].

6.5. Address Block TLV Types

A new registry for Message-Type-independent Address Block TLV Types has been created, with initial assignments and allocation policies as specified in Table 10.

Type	Description	Allocation Policy
0-127	Unassigned	Expert Review
128-223	Message-Type-specific	Reserved, see Table 11
224-255	Unassigned	Experimental Use

Table 10: Address Block TLV Types

Address Block TLV Types 128-223 are reserved for Message-Type-specific Address Block TLVs, for which a new registry is created with the registration of a Message Type, and with initial assignments and allocation policies as specified in Table 11.

Type	Description	Allocation Policy
0-127	Common to all Message Types	Reserved
128-223	Message-Type-specific	See Below
224-255	Common to all Message Types	Reserved

Table 11: Message-Type-specific Address Block TLV Types

Allocation policies for Message-Type-specific Address Block TLV Types MUST be specified when creating the registry associated with the containing Message Type, see Section 6.2.1.

6.5.1. Address Block TLV Type Extension Registry Creation

When an Address Block TLV Type is registered, then a new registry for type extensions of that type must be created. A document that defines a Message TLV Type MUST also specify the mechanism by which its type extensions are allocated, from among those in [BCP26].

7. Security Considerations

This specification does not describe a protocol; it describes a packet format. As such, it does not specify any security considerations; these are matters for a protocol using this specification. However, some security mechanisms are enabled by this specification and may form part of a protocol using this specification. Mechanisms that may form part of an authentication and integrity approach in a protocol using this specification are described in Section 7.1. Mechanisms that may form part of a confidentiality approach in a protocol using this specification are described in Section 7.2. There is, however, no requirement that a protocol using this specification should use either.

7.1. Authentication and Integrity Suggestions

The authentication and integrity suggestions made here are based on the intended usage in Appendix B, specifically that:

- o Messages are designed to be carriers of protocol information and MAY, at each hop, be forwarded and/or processed by the protocol using this specification.

- o Packets are designed to carry a number of messages between neighboring MANET routers in a single transmission and over a single logical hop.

Consequently:

- o For forwarded messages where the message is unchanged by forwarding MANET routers, end-to-end authentication and integrity MAY be implemented, between MANET routers with an existing security association, by including a suitable Message TLV containing a cryptographic signature in the message. Since <msg-hop-count> and <msg-hop-limit> are the only fields that should be modified when such a message is forwarded in this manner, this signature can be calculated based on the entire message, including the Message Header, with the <msg-hop-count> and <msg-hop-limit> fields set to 0, if present.

- o Hop-by-hop packet level authentication and integrity MAY be implemented, between MANET routers with an existing security association, by including a suitable Packet TLV containing a cryptographic signature to the packet. Since packets are received as transmitted, this signature can be calculated based on the entire packet or on parts thereof as appropriate.

7.2. Confidentiality Suggestions

This specification does not explicitly enable protecting packet/ message confidentiality. Such confidentiality would normally, when required, be provided hop-by-hop, either by link-layer mechanisms or at the IP layer using [RFC4301], and would apply to a packet only.

It is possible, however, for a protocol using this specification to protect the confidentiality of information included in a Packet, Message, or Address Block TLV by specifying that the Value field of that TLV Type be encrypted, as well as specifying the encryption mechanism.

In an extreme case, all information can be encrypted by defining either:

- o A packet, consisting of only a Packet Header (with no messages) and containing a Packet TLV, where the Packet TLV Type indicates that its Value field contains one or more encrypted messages. Upon receipt, and once this Packet TLV is successfully decrypted, these messages may then be parsed according to this specification and processed according to the protocol using this specification.

- o A message, consisting of only a Message Header and a single Message TLV, where the Message TLV Type indicates that its Value field contains an encrypted version of the message’s remaining Message TLVs, Address Blocks, and Address Block TLVs. Upon receipt, and once this Message TLV is successfully decrypted, the

complete message may then be parsed according to this specification and processed according to the protocol using this specification.

In either case, the protocol MUST define the encrypted TLV Type, as well as the format of the encrypted data block contained in the Value field of the TLV.

8. Contributors

This specification is the result of the joint efforts of the following contributors from the OLSRv2 Design Team, listed alphabetically:

- o Cedric Adjih, INRIA, France, <Cedric.Adjih@inria.fr>
- o Emmanuel Baccelli, INRIA, France, <Emmanuel.Baccelli@inria.fr>
- o Thomas Heide Clausen, LIX, Ecole Polytechnique, France, <T.Clausen@computer.org>
- o Justin W. Dean, NRL, USA, <jdean@itd.nrl.navy.mil>
- o Christopher Dearlove, BAE Systems, UK, <chris.dearlove@baesystems.com>
- o Satoh Hiroki, Hitachi SDL, Japan, <hiroki.satoh.yj@hitachi.com>
- o Philippe Jacquet, INRIA, France, <Philippe.Jacquet@inria.fr>
- o Monden Kazuya, Hitachi SDL, Japan, <kazuya.monden.vw@hitachi.com>

9. Acknowledgments

The authors would like to acknowledge the team behind OLSR [RFC3626], including Anis Laouiti (INT, France), Pascale Minet, Laurent Viennot (both at INRIA, France), and Amir Qayyum (Center for Advanced Research in Engineering, Pakistan) for their contributions. Elwyn Davies (Folly Consulting, UK), Lars Eggert (Nokia, Finland), Chris Newman (Sun Microsystems, USA), Tim Polk (NIST, USA), and Magnus Westerlund (Ericsson, Sweden) all provided detailed reviews and insightful comments.

The authors would like to gratefully acknowledge the following people for intense technical discussions, early reviews, and comments on the specification and its components (listed alphabetically):

- o Brian Adamson (NRL)
- o Teco Boot (Infinity Networks)
- o Florent Brunneau (LIX)
- o Ian Chakeres (CenGen)
- o Alan Cullen (BAE Systems)
- o Ulrich Herberg (LIX)
- o Joe Macker (NRL)
- o Yasunori Owada (Niigata University)
- o Charlie E. Perkins (WiChorus)
- o Henning Rogge (FGAN)
- o Andreas Schjonhaug (LIX)

and the entire IETF MANET working group.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[BCP26] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[SingleUNIX] IEEE Std 1003.1, The Open Group, and ISO/IEC JTC 1/SC22/WG15, "Single UNIX Specification, Version 3, 2004 Edition", April 2004.

10.2. Informative References

[RFC3626] Clausen, T. and P. Jacquet, "The Optimized Link State Routing Protocol", RFC 3626, October 2003.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[Stevens] Stevens, W., "TCP/IP Illustrated Volume 1 - The Protocols", 1994.

Appendix A. Multiplexing and Demultiplexing

The packet and message format specified in this document is designed to allow zero or more messages to be contained within a single packet. Such messages may be from the same or different protocols. Thus, a multiplexing and demultiplexing process MUST be present.

Multiplexing messages on a given MANET router into a single packet, rather than having each message generate its own packet, reduces the total number of octets and the number of packets transmitted by that MANET router.

The multiplexing and demultiplexing process running on a given UDP port or IP protocol number, and its associated protocols, MUST:

- For each Message Type, a protocol -- unless specified otherwise, the one making the IANA reservation for that Message Type -- MUST be designated as the "owner" of that Message Type.

- The Packet Header fields, including the Packet TLV Block, are used by the multiplexing and demultiplexing process, which MAY make such information available for use in its protocol instances.

- The <pkt-seq-num> field, if present, contains a sequence number that is incremented by 1 for each packet generated by a node. The sequence number after 65535 is 0. In other words, the sequence number "wraps" in the usual way.

- Incoming messages MUST be either silently discarded or MUST be delivered to the instance of the protocol that owns the associated Message Type. Incoming messages SHOULD NOT be delivered to any other protocol instances and SHOULD NOT be delivered to more than one protocol instance.

- Outgoing messages of a given type MUST be generated only by the protocol instance that owns that Message Type and be delivered to the multiplexing and demultiplexing process.

- If two protocols both wish to use the same Message Type, then this interaction SHOULD be specified by the protocol that is the designated owner of that Message Type.

Appendix B. Intended Usage

This appendix describes the intended usage of Message Header fields, including their content and use. Alternative uses of this specification are permitted.

The message format specified in this document is designed to carry MANET routing protocol signaling between MANET routers and to support scope-limited flooding as well as point-to-point delivery.

Messages are designed to be able to be forwarded over one or more logical hops, in a new packet for each logical hop. Each logical hop may consist of one or more IP hops.

Specifically, scope-limited flooding is supported for messages when:

- The <msg-orig-addr> field, if present, contains the unique identifier of the MANET router that originated the message.

- The <msg-seq-num> field, if present, contains a sequence number that starts at 0 when the first message of a given type is generated by the originator node, and that is incremented by 1 for each message generated of that type. The sequence number after 65535 is 0. In other words, the sequence number "wraps" in the usual way.

- If the <msg-orig-addr> and <msg-seq-num> fields are both present, then the Message Header provides for duplicate suppression, using the identifier consisting of the message's <msg-orig-addr>, <msg-seq-num>, and <msg-type>. These serve to uniquely identify the message in the MANET within the time period until <msg-seq-num> is repeated, i.e., wraps around to a matching value.

- <msg-hop-limit> field, if present, contains the number of hops on which the packet is allowed to travel before being discarded by a MANET router. The <msg-hop-limit> is set by the message originator and is used to prevent messages from endlessly circulating in a MANET. When forwarding a message, a MANET router should decrease the <msg-hop-limit> by 1, and the message should be discarded when <msg-hop-limit> reaches 0.

- <msg-hop-count> field, if present, contains the number of hops on which the packet has traveled across the MANET. The <msg-hop-count> is set to 0 by the message originator and is used to prevent messages from endlessly circulating in a MANET. When forwarding a message, a MANET router should increase <msg-hop-count> by 1 and should discard the message when <msg-hop-count> reaches 255.

- If the <msg-hop-limit> and <msg-hop-count> fields are both present, then the Message Header provides the information to make forwarding decisions for scope-limited flooding. This may be by any appropriate flooding mechanism specified by a protocol using this specification.

Appendix C. Examples

This appendix contains some examples of parts of this specification.

C.1. Address Block Examples

The following examples illustrate how some combinations of addresses may be efficiently included in Address Blocks. These examples are for IPv4, with address-length equal to 4. a, b, c, etc. represent distinct, non-zero octet values.

Note that it is permissible to use a less efficient representation, in particular one in which the ahashead and ahasfulltail flags are cleared ('0'), and hence head-length = 0, tail-length = 0, mid-length = address-length, and (with no address prefixes) the Address Block consists of the number of addresses, <addr-flags> with value 0, and a list of the unaggregated addresses. This is the most efficient way to represent a single address, and the only way to represent, for example, a.b.c.d and e.f.g.h in one Address Block.

Examples:

- To include a.b.c.d, a.b.e.f, and a.b.g.h:

  * head-length = 2;

  * tail-length = 0;

  * mid-length = 2;

  * <addr-flags> has ahashead set (value 128);

  * <tail-length> and <tail> are omitted.

  The Address Block is then 3 128 2 a b c d e f g h (11 octets).

- To include a.b.c.g and d.e.f.g:

  * head-length = 0;

  * tail-length = 1;

  * mid-length = 3;

  * <addr-flags> has ahasfulltail set (value 64);

  * <head-length> and <head> are omitted.

The Address Block is then 2 64 1 g a b c d e f (10 octets).

o To include a.b.d.e and a.c.d.e:

* head-length = 1;
* tail-length = 2;
* mid-length = 1;
* <addr-flags> has ahashead and ahasfulltail set (value 192).

The Address Block is then 2 192 1 a 2 d e b c (9 octets).

o To include a.b.0.0, a.c.0.0, and a.d.0.0:

* head-length = 1;
* tail-length = 2;
* mid-length = 1;
* <addr-flags> has ahashead and ahaszerotail set (value 160);
* <tail> is omitted.

The Address Block is then 3 160 1 a 2 b c d (8 octets).

o To include a.b.0.0 and c.d.0.0:

* head-length = 0;
* tail-length = 2;
* mid-length = 2;
* <addr-flags> has ahaszerotail set (value 32);
* <head> and <tail> are omitted.

The Address Block is then 2 32 2 a b c d (7 octets).

o To include a.b.0.0/n and c.d.0.0/n:

* head-length = 0;
* tail-length = 2;

* mid-length = 2;

* <addr-flags> has ahaszerotail and ahassingleprelen set (value 48);

* <head> and <tail> are omitted.

The Address Block is then 2 48 2 a b c d n (8 octets).

o To include a.b.0.0/n and c.d.0.0/m:

* head-length = 0;

* tail-length = 2;

* mid-length = 2;

* <addr-flags> has ahaszerotail and ahasmultiprelen set (value 40);

* <head> and <tail> are omitted.

The Address Block is then 2 40 2 a b c d n m (9 octets).

C.2. TLV Examples

Assume the definition of an Address Block TLV with type EXAMPLE1 (and no type extension) that has single octet values per address. There are a number of ways in which values a, a, b, and c may be associated with the four addresses in the preceding Address Block, where c is a default value that can be omitted.

Examples:

o Using one multivalue TLV to cover all of the addresses:

* <tlv-flags> has thasvalue and tismultivalue set (value 20);

* <index-start> and <index-stop> are omitted;

* <length> = 4 (single-length = 1).

* The TLV is then EXAMPLE1 20 4 a a b c (7 octets).

o Using one multivalue TLV and omitting the last address:

* <tlv-flags> has thasmultiindex, thasvalue, and tismultivalue set (value 52);

* <index-start> = 0;

* <index-stop> = 2;

* <length> = 3 (single-length = 1).

* The TLV is then EXAMPLE1 52 0 2 3 a a b (8 octets).

o Using two single value TLVs and omitting the last address. First:

* <tlv-flags> has thasmultiindex and thasvalue set (value 48);

* <index-start> = 0;

* <index-stop> = 1;

* <length> = 1;

* <value> = a.

* The TLV is then EXAMPLE1 48 0 1 1 a (6 octets).

Second:

* <tlv-flags> has thassingleindex and thasvalue set (value 80);

* <index-start> = 2;

* <index-stop> is omitted;

* <length> = 1;

* <value> = b.

* The TLV is then EXAMPLE1 80 2 1 b (5 octets).

Total length of TLVs is 11 octets.

In this case, the first of these is the most efficient. In other cases, patterns such as the others may be preferred. Regardless of efficiency, any of these may be used.

Assume the definition of an Address Block TLV with type EXAMPLE2 (and no type extension) that has no value and that is to be associated with the second and third addresses in an Address Block. This can be indicated with a single TLV:

- <tlv-flags> has thasmultiindex set (value 32);
- <index-start> = 1;
- <index-stop> = 2;
- <length> and <value> are omitted.
- The TLV is then EXAMPLE2 32 1 2 (4 octets).

Assume the definition of a Message TLV with type EXAMPLE3 (and no type extension) that can take a Value field of any length. For such a TLV with 8 octets of data (a to h):

- <tlv-flags> has thasvalue set (value 16);
- <index-start> and <index-stop> are omitted;
- <length> = 8.
- The TLV is then EXAMPLE3 16 8 a b c d e f g h (11 octets).

If, in this example, the number of data octets were 256 or greater, then <tlv-flags> would also have thasextlen set and have value 24. The length would require two octets (most significant first). The TLV length would be 4 + N octets, where N is the number of data octets (it can be 3 + N octets if N is 255 or less).

Appendix D. Illustrations

This informative appendix illustrates the elements that are normatively specified in Section 5.

Bits labeled Rsv should be cleared ('0'). Bits labeled M may be cleared ('0') or set ('1').

D.1. Packet

This section illustrates possible options for the <packet> element. These are differentiated by the flags field in the first octet. The packet may include any number (zero or more) of messages. The number of messages is determined from when the packet is exhausted, given the packet length from the network layer.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0|0|0|0|0|0|Rsv|                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                            Message                            |
    |                                                               |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                            Message                            |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0|0|0|0|1|0|Rsv|    Packet Sequence Number     |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                            Message                            |
    |                                                               |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                            Message                            |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0|0|0|0|0|1|Rsv|                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    |                        Packet TLV Block                       |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                             Message                           |
    |                                                               |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                               ...                             :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                             Message                           |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0|0|0|0|1|1|Rsv|    Packet Sequence Number     |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
   |                                                               |
   |                       Packet TLV Block                        |
   |                                                               |
   |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
   |                                                               |
   |                            Message                            |
   |                                                               |
   |                                               +-+-+-+-+-+-+-+-+
   |                                               |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
   |                                                               |
   :                              ...                              :
   |                                                               |
   |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
   |                                                               |
   |                            Message                            |
   |                                               +-+-+-+-+-+-+-+-+
   |                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D.2. Message

This section illustrates possible options for the <message> element. These are differentiated by their second (flags) octet. The length of the Message Body is determined using the Message Size field, which is the combined length of all the fields shown. The field labeled MAL defines the length of all addresses (including the Originator Address, if present, and all addresses in Address Blocks) in octets, as one more than the value in the field.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  |0|0|0|0|  MAL  |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Message Body                          |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  |1|0|0|0|  MAL  |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Message Body                          |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  |0|1|0|0|  MAL  |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Hop Limit   |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                         Message Body                          |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |1|1|0|0|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |0|0|1|0|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Count   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |1|0|1|0|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Count   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |0|1|1|0|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                          Message Body                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |1|1|1|0|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                          Message Body                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |0|0|0|1|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Message Sequence Number    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                          Message Body                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Message Type  |1|0|0|1|  MAL  |         Message Size          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                      Originator Address                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    Message Sequence Number    |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
     |                                                               |
     |                         Message Body                          |
     |                                                               |
     |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Message Type  |0|1|0|1|  MAL  |         Message Size          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |   Hop Limit   |    Message Sequence Number    |               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
     |                                                               |
     |                         Message Body                          |
     |                                                               |
     |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Message Type  |1|1|0|1|  MAL  |         Message Size          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                      Originator Address                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |   Hop Limit   |    Message Sequence Number    |               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
     |                                                               |
     |                         Message Body                          |
     |                                                               |
     |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |0|0|1|1|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Count   |    Message Sequence Number    |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                          Message Body                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |1|0|1|1|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Count   |    Message Sequence Number    |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                          Message Body                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |0|1|1|1|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |    Message Sequence Number    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                          Message Body                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |1|1|1|1|  MAL  |         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |    Message Sequence Number    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D.3. Message Body

This section illustrates the format of a Message Body (the <message> element excluding its initial <msg-header> element). The Message Body includes one Message TLV Block (containing zero or more TLVs) and may include any number (zero or more) of Address Block and Address Block TLV Block pairs.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                       Message TLV Block                       |
     |                                               +-+-+-+-+-+-+-+-+
     |                                               |               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
     |                                                               |
     |                         Address Block                         |
     |                                                               |
     |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |               |                                               |
     +-+-+-+-+-+-+-+-+                                               |
     |                    Address Block TLV Block                    |
     |                                                               |
     |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
     |                                                               |
     :                              ...                              :
     |                                                               |
     |                                               +-+-+-+-+-+-+-+-+
     |                                               |               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
     |                                                               |
     |                         Address Block                         |
     |                                                               |
     |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
     |                                                               |
     |                    Address Block TLV Block                    |
     |                                                               |
     |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D.4. Address Block

This section illustrates possible options for the <addr-block> element. These are differentiated by their second (flags) octet. The number of Mid elements is equal to the number of addresses (except when mid-length is zero, when there are no Mid elements). Where a variable number of prefix length fields is shown, their number is equal to the number of addresses.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number Addrs  |0|0|0|0|0| Rsv |              Mid              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Mid (cont)           |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
:                              ...                              :
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |              Mid              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Mid (cont)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number Addrs  |1|0|0|0|0| Rsv |  Head Length  |     Head      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Head (cont)  |              Mid              |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
:                              ...                              :
|                                                               |
|               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |              Mid              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|1|0|0|0| Rsv |  Tail Length  |     Tail      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Tail (cont)  |              Mid              |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|1|0|0|0| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |  Tail Length  |             Tail              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|0|1|0|0| Rsv |  Tail Length  |      Mid      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Mid (cont)   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|0|1|0|0| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |  Tail Length  |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|0|0|1|0| Rsv |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Mid (cont)           |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Mid (cont)           | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|0|0|1|0| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |              Mid              |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|1|0|1|0| Rsv |  Tail Length  |     Tail      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Tail (cont)  |              Mid              |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|1|0|1|0| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |  Tail Length  |             Tail              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Prefix Length |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|0|1|1|0| Rsv |  Tail Length  |      Mid      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Mid (cont)   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|0|1|1|0| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |  Tail Length  |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|0|0|0|1| Rsv |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Mid (cont)           |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Mid (cont)           | Prefix Length |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|0|0|0|1| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |              Mid              |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|1|0|0|1| Rsv |  Tail Length  |     Tail      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Tail (cont)  |              Mid              |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|1|0|0|1| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |  Tail Length  |             Tail              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Prefix Length |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    :                              ...                              :
    |                                               +-+-+-+-+-+-+-+-+
    |                                               | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0|0|1|0|1| Rsv |  Tail Length  |      Mid      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Mid (cont)   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1|0|1|0|1| Rsv |  Head Length  |     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |  Tail Length  |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              | Prefix Length |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               | Prefix Length |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D.5. TLV Block

This section illustrates the format of a <tlv-block> element. There may be any number (zero or more) of TLVs, with total length of the TLVs (in octets) equal to the Length field.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Length             |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                              TLV                              |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                              TLV                              |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D.6. TLV

This section illustrates possible options for the <tlv> element. These are differentiated by their second (flags) octet. If there are no Index fields, then this may be a Packet TLV, Message TLV, or Address Block TLV; if there are one or two Index fields, then this must be an Address Block TLV. The Length field gives the length of the Value field (in octets).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |0|0|0|0|0|0|Rsv|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |1|0|0|0|0|0|Rsv|   Type Ext    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |0|1|0|0|0|0|Rsv|  Index Start  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |1|1|0|0|0|0|Rsv|   Type Ext    |  Index Start  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |0|0|1|0|0|0|Rsv|  Index Start  |  Index Stop   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |1|0|1|0|0|0|Rsv|   Type Ext    |  Index Start  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Index Stop   |
+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |0|0|0|1|0|M|Rsv|    Length     |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |1|0|0|1|0|M|Rsv|   Type Ext    |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |0|1|0|1|0|0|Rsv|  Index Start  |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |1|1|0|1|0|0|Rsv|   Type Ext    |  Index Start  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Length     |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |0|0|1|1|0|M|Rsv|  Index Start  |  Index Stop   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Length     |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                              Value                            |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |1|0|1|1|0|M|Rsv|   Type Ext    |  Index Start  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Index Stop   |    Length     |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                              Value                            |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |0|0|0|0|1|M|Rsv|            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                              Value                            |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |1|0|0|1|1|M|Rsv|   Type Ext    |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Length (cont) |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |0|1|0|1|1|0|Rsv|  Index Start  |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Length (cont) |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |1|1|0|1|1|0|Rsv|   Type Ext    |  Index Start  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Length             |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |0|0|1|1|1|M|Rsv|  Index Start  |  Index Stop   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Length             |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+

     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |1|0|1|1|1|M|Rsv|   Type Ext    |  Index Start  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Index Stop   |            Length             |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

Appendix E. Complete Example

The following example packet is included with the intent to exemplify how Packet Headers and Message Headers are constructed, and how addresses and attributes are encoded using Address Blocks and TLV Blocks. This example is specifically not constructed to exhibit maximum message or packet size reduction.

The Packet Header has the phasseqnum flag of its flags field set (value 8), and hence has a Packet Sequence Number, but no Packet TLV Block.

The packet contains a single message with length 54 octets. This message has the mhasorig, mhashoplimit, mhashopcount, and mhasseqnum flags of its four-bit flags field set (value 15), and hence includes an Originator Address, a Hop Limit, a Hop Count, and a Message Sequence Number (which is type independent). Its four-bit Message Address Length field has value 3, and hence addresses in the message have length four octets, here being IPv4 addresses. The message has a Message TLV Block with content length 9 octets, containing a single

Message TLV. This TLV has the thasvalue flag of its flags octet set (value 16), and hence contains a Value field, with preceding Value Length 6 octets. The message then has two Address Blocks, each with a following Address Block TLV Block.

The first Address Block contains 2 address prefixes. It has the ahaszerotail and ahassingleprelen flags of its flags octet set (value 48), and hence has no Head (head-length is zero octets). It has a tail-length of 2 octets, hence mid-length is two octets. The two Tail octets of each address are not included (since ahaszerotail is set) and have value zero. The Address Block has a single prefix length. The following Address Block TLV Block is empty (content length 0 octets).

The second Address Block contains 3 addresses. It has the ahashead flag of its flags octet set (value 128), has head-length 2 octets, and no Tail (tail-length is zero octets); hence, mid-length is two octets. It is followed by an Address Block TLV Block, with content length 9 octets, containing two Address Block TLVs. The first of these TLVs has the thasvalue flag of its flags octet set (value 16), and has a single value of length 2 octets, which applies to all of the addresses in the Address Block. The second of these TLVs has the thasmultiindex flag of its flags octet set (value 32), and hence has no Value Length or Value fields. It has two Index fields (Index Start and Index Stop), which indicate those addresses this TLV applies to (inclusive range, counting from zero).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 1 0 0 0|    Packet Sequence Number     | Message Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 0 0 1 1|0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0|   Orig Addr   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Originator Address (cont)           |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Hop Count   |    Message Sequence Number    |0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 1 0 0 1|   TLV Type    |0 0 0 1 0 0 0 0|0 0 0 0 0 1 1 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Value (cont)          |0 0 0 0 0 0 1 0|0 0 1 1 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 1 0|              Mid              |      Mid      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Mid (cont)   | Prefix Length |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 1 1|1 0 0 0 0 0 0 0|0 0 0 0 0 0 1 0|     Head      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Head (cont)  |              Mid              |      Mid      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Mid (cont)   |              Mid              |0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 1 0 0 1|   TLV Type    |0 0 0 1 0 0 0 0|0 0 0 0 0 0 1 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Value             |   TLV Type    |0 0 1 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Index Start  |  Index Stop   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Authors' Addresses

Thomas Heide Clausen
LIX, Ecole Polytechnique

Phone: +33 6 6058 9349
EMail: T.Clausen@computer.org
URI: http://www.thomasclausen.org/

Christopher M. Dearlove
BAE Systems ATC

Phone: +44 1245 242194
EMail: chris.dearlove@baesystems.com
URI: http://www.baesystems.com/

Justin W. Dean
Naval Research Laboratory

Phone: +1 202 767 3397
EMail: jdean@itd.nrl.navy.mil
URI: http://pf.itd.nrl.navy.mil/

Cedric Adjih
INRIA Rocquencourt

Phone: +33 1 3963 5215
EMail: Cedric.Adjih@inria.fr
URI: http://menetou.inria.fr/~adjih/